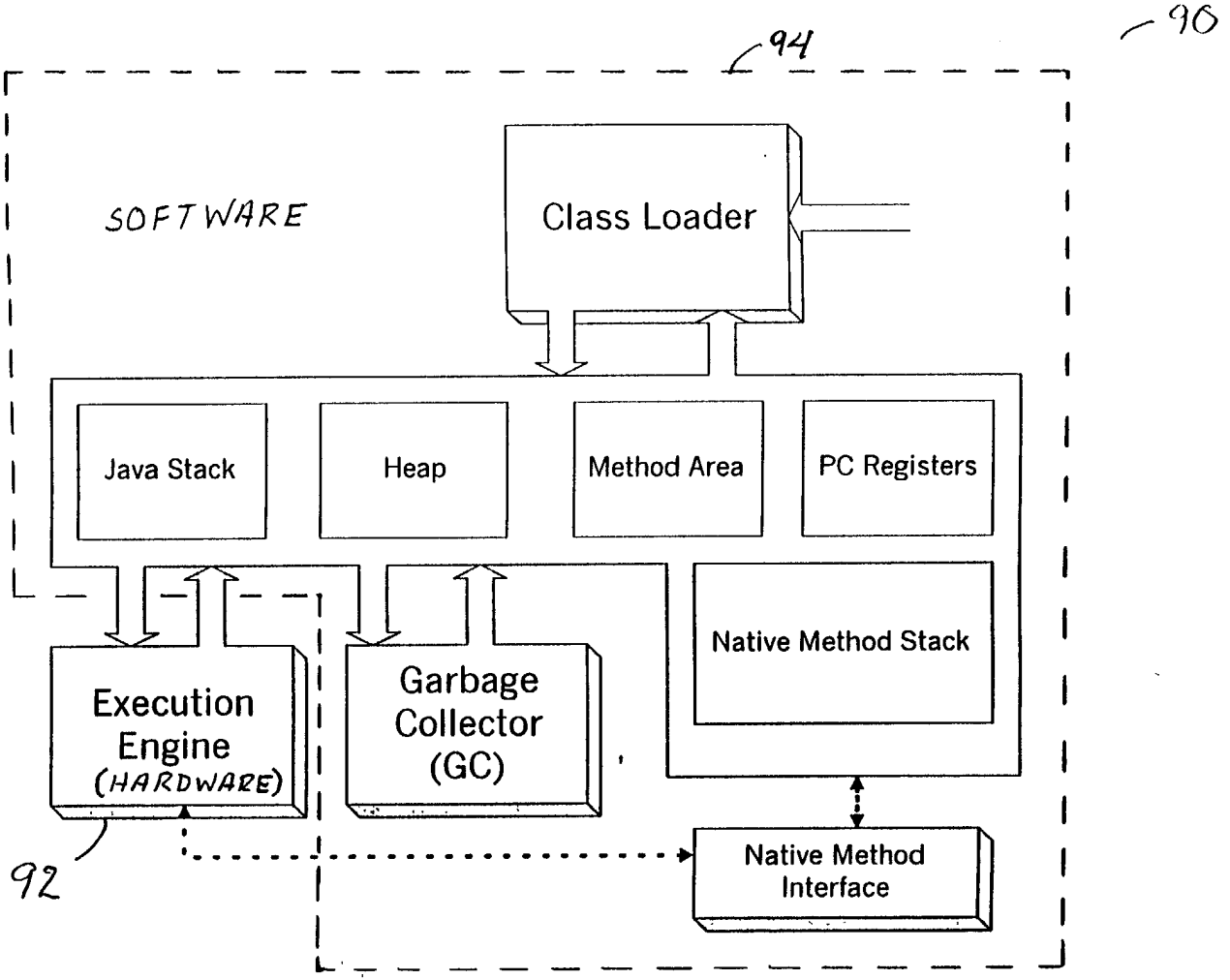

90
94
SOFTWARE
Class Loader
Java Stack
Heap
Method Area
PC Registers
Native Method Stack
Execution Engine (HARDWARE)
Garbage Collector (GC)
Native Method Interface
92

FIG. 1

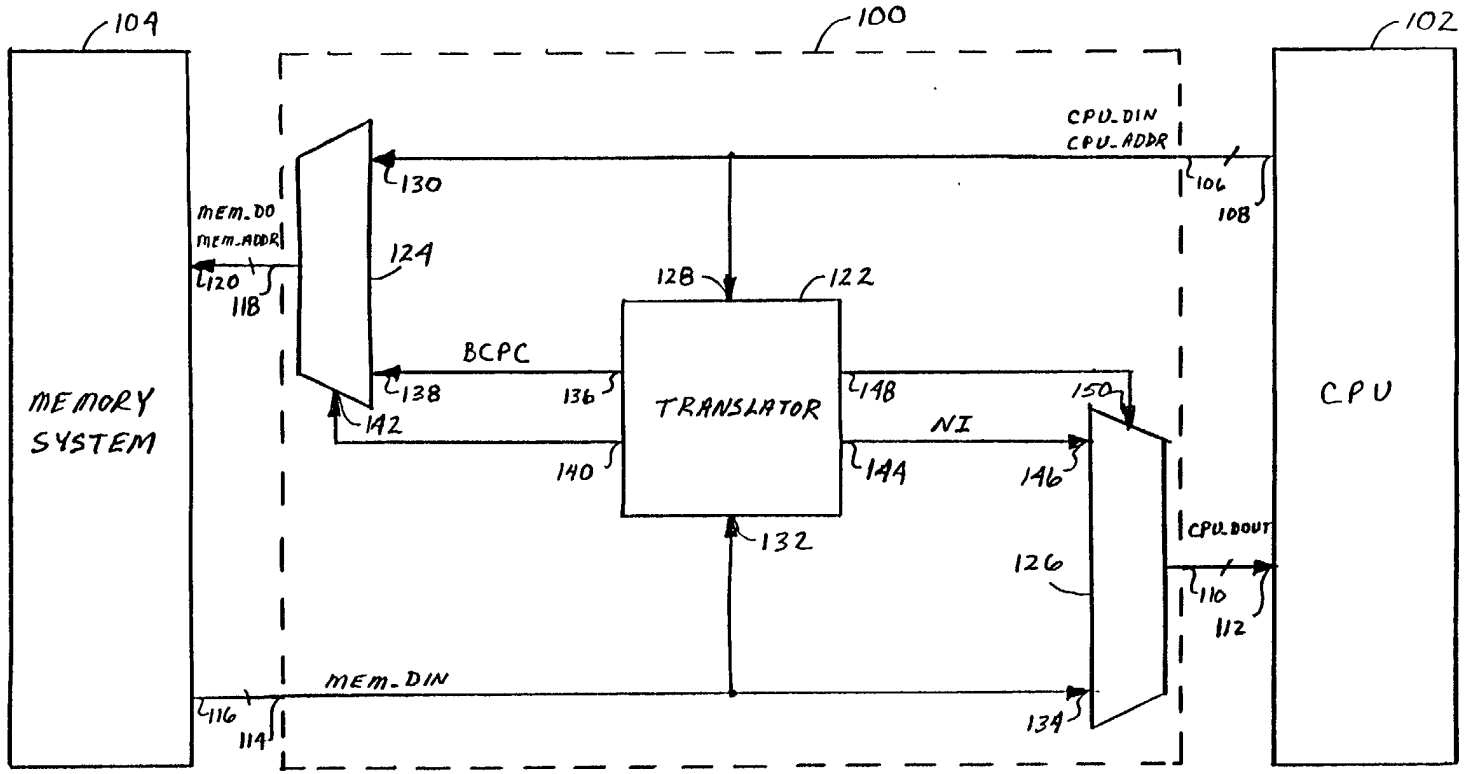

104
100
102
MEMORY SYSTEM
CPU
TRANSLATOR
CPU_DIN
CPU_ADDR
106
108
130
MEM_DO
MEM_ADDR
120
118
124
128
122
BCPC
138
136
142
140
148
144
NI
150
146
132
126
CPU_DOUT
110
112
MEM_DIN
116
114
134

FIG. 2

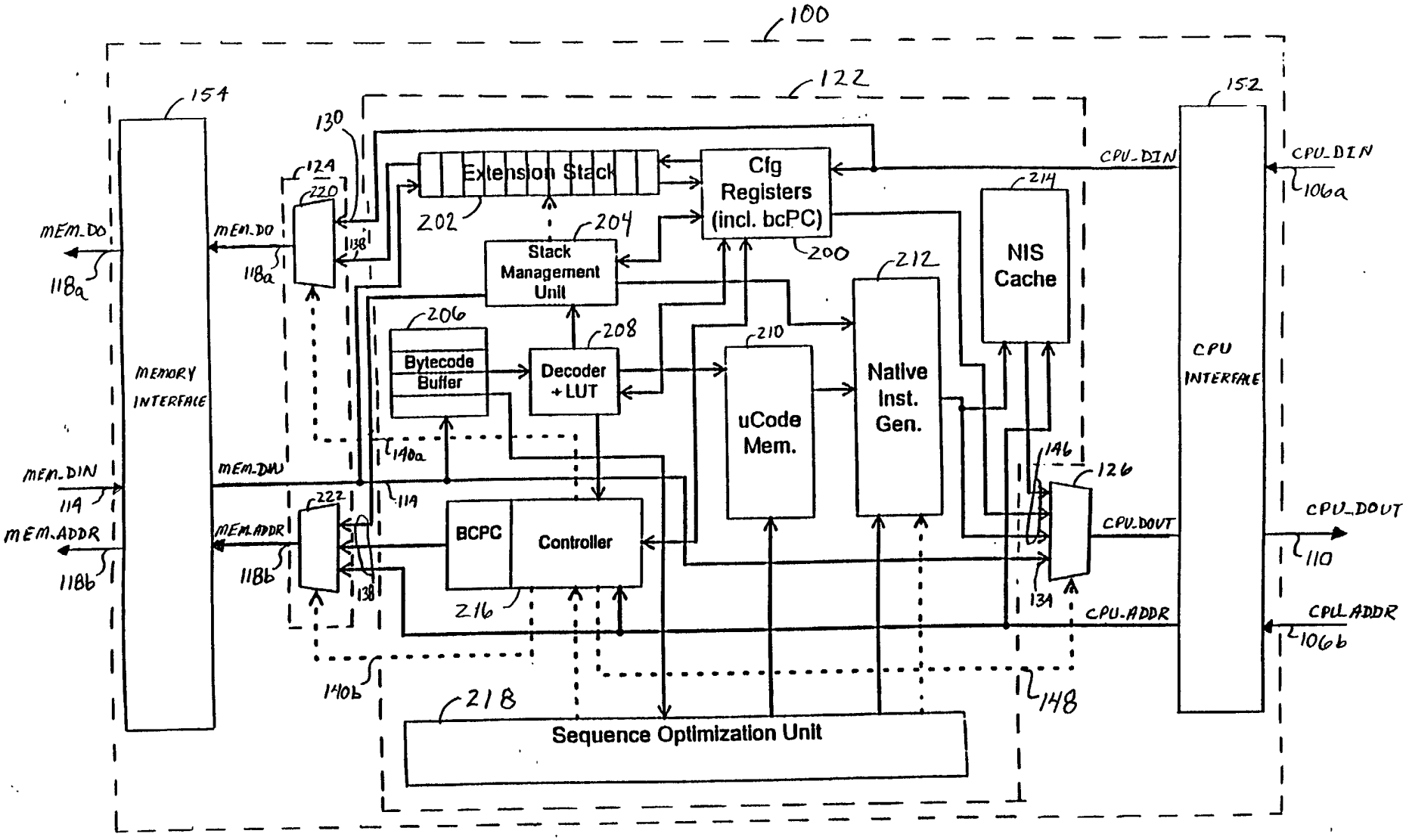

100
122
154
152
130
124
220
Extension Stack
202
Cfg Registers (incl. bcPC)
200
CPU_DIN
CPU_DIN
106a
214
NIS Cache
204
Stack Management Unit
212
MEM_DO
118a
MEM_DO
138
206
Bytecode Buffer
208
Decoder + LUT
210
uCode Mem.
Native Inst. Gen.
MEMORY INTERFACE
CPU INTERFACE
MEM_DIN
119
MEM_DIN
140a
146
126
222
MEM_ADDR
118b
MEM_ADDR
BCPC
Controller
CPU_DOUT
CPU_DOUT
110
134
CPU_ADDR
CPU_ADDR
106b
216
140b
148
218
Sequence Optimization Unit

FIG. 3

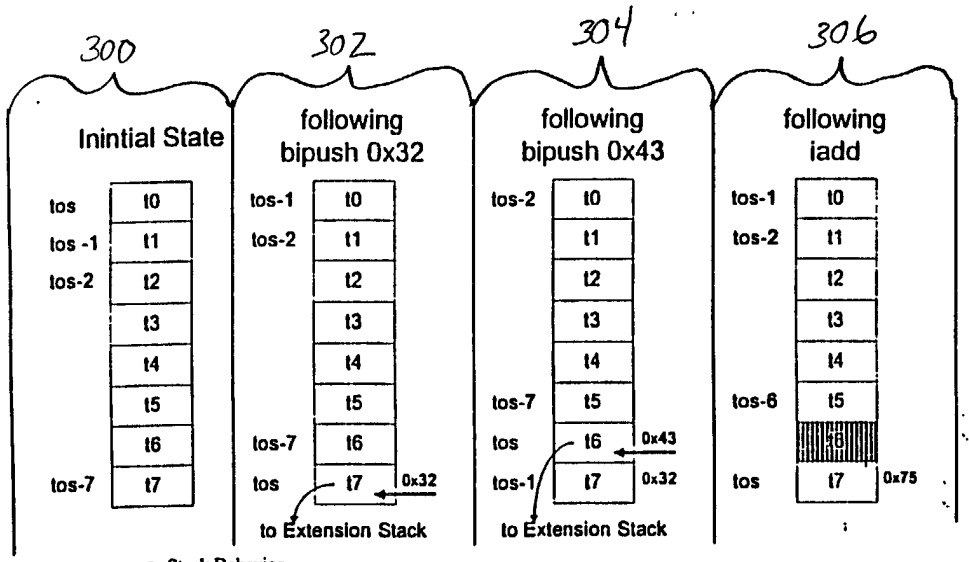


Figure 3 Stack Behavior

FIG. 4

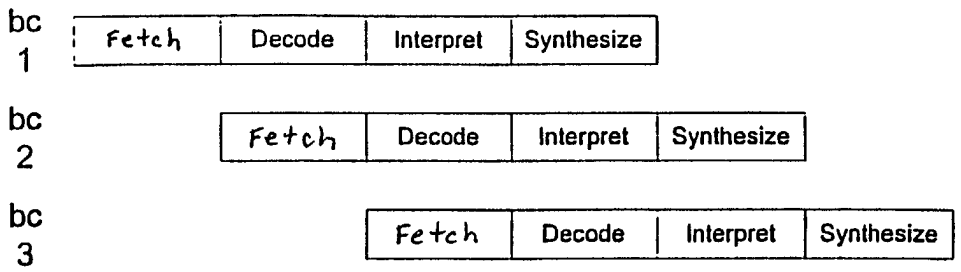
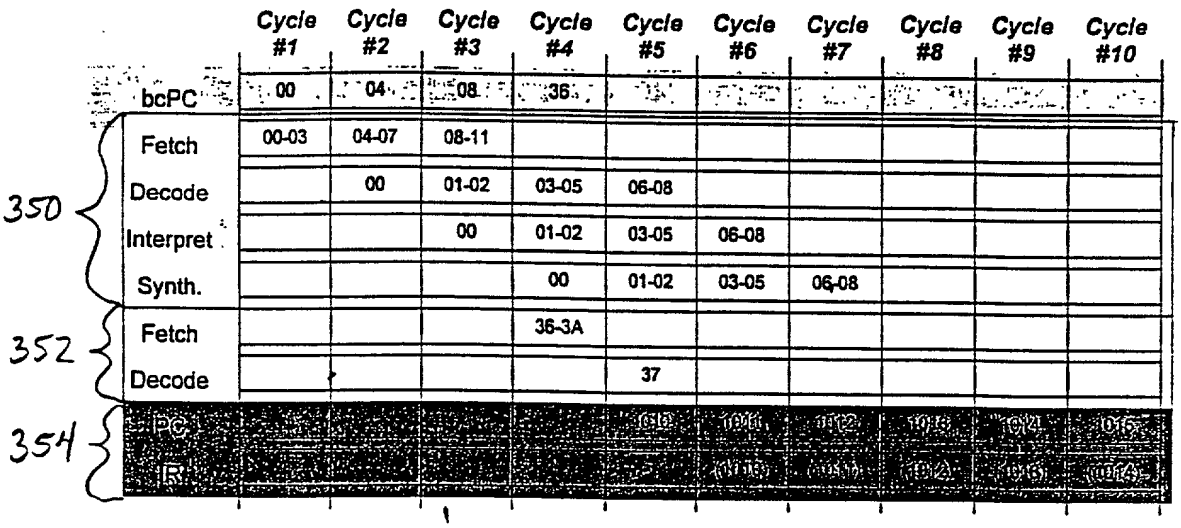

bc 1
Fetch
Decode
Interpret
Synthesize
bc 2
Fetch
Decode
Interpret
Synthesize
bc 3
Fetch
Decode
Interpret
Synthesize

FIG. 5



		Cycle #1	Cycle #2	Cycle #3	Cycle #4	Cycle #5	Cycle #6	Cycle #7	Cycle #8	Cycle #9	Cycle #10
	bcPC	00	04	08	36						
350	Fetch	00-03	04-07	08-11							
	Decode		00	01-02	03-05	06-08					
	Interpret			00	01-02	03-05	06-08				
	Synth.				00	01-02	03-05	06-08			
352	Fetch				36-3A						
	Decode					37					
354	PC					010	011	012	013	014	015
	IR						0010	0011	0012	0013	0014

FIG. 6

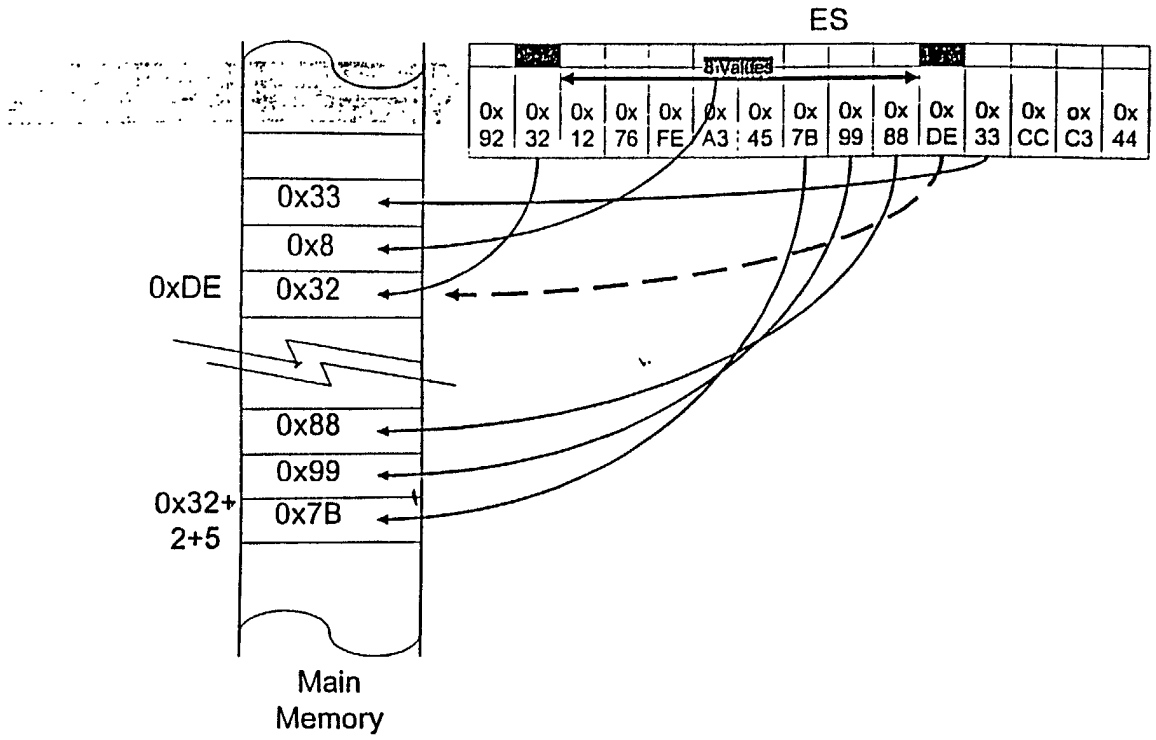

ES
8 Values
0x 92
0x 32
0x 12
0x 76
0x FE
0x A3
0x 45
0x 7B
0x 99
0x 88
0x DE
0x 33
0x CC
ox C3
0x 44
0x33
0x8
0xDE
0x32
0x88
0x99
0x32+ 2+5
0x7B
Main Memory

FIG. 7

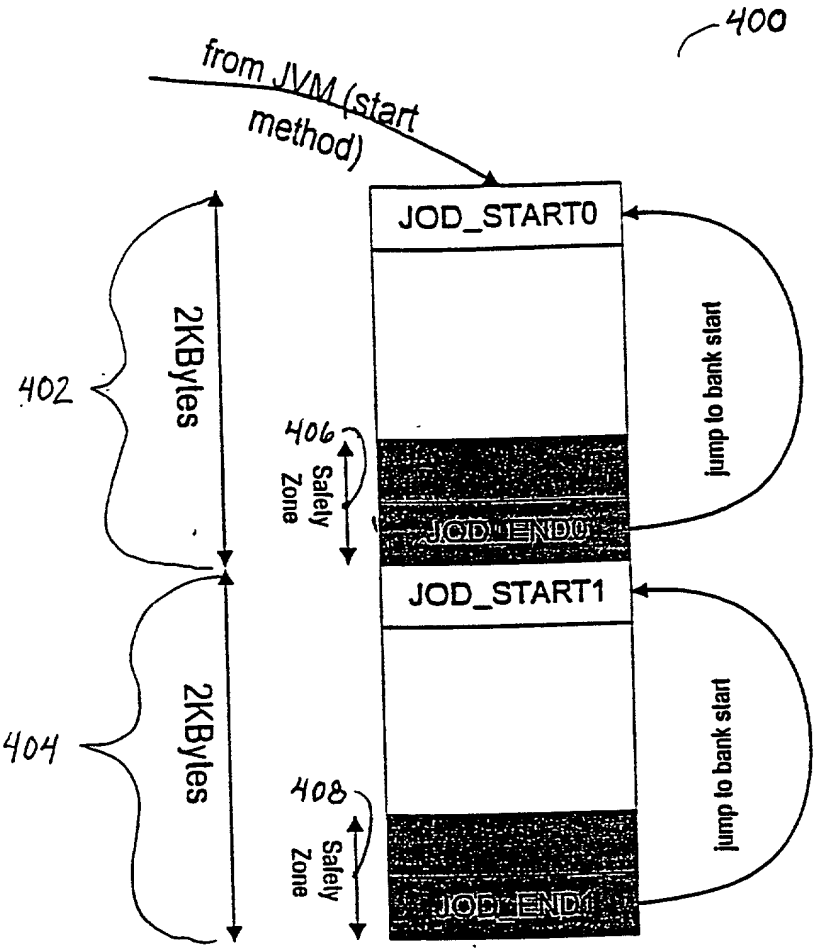

400
from JVM (start method)
JOD_START0
2KBytes
402
406
Safety Zone
JOD_END0
jump to bank start
JOD_START1
2KBytes
404
408
Safety Zone
JOD_END1
jump to bank start

FIG. 8

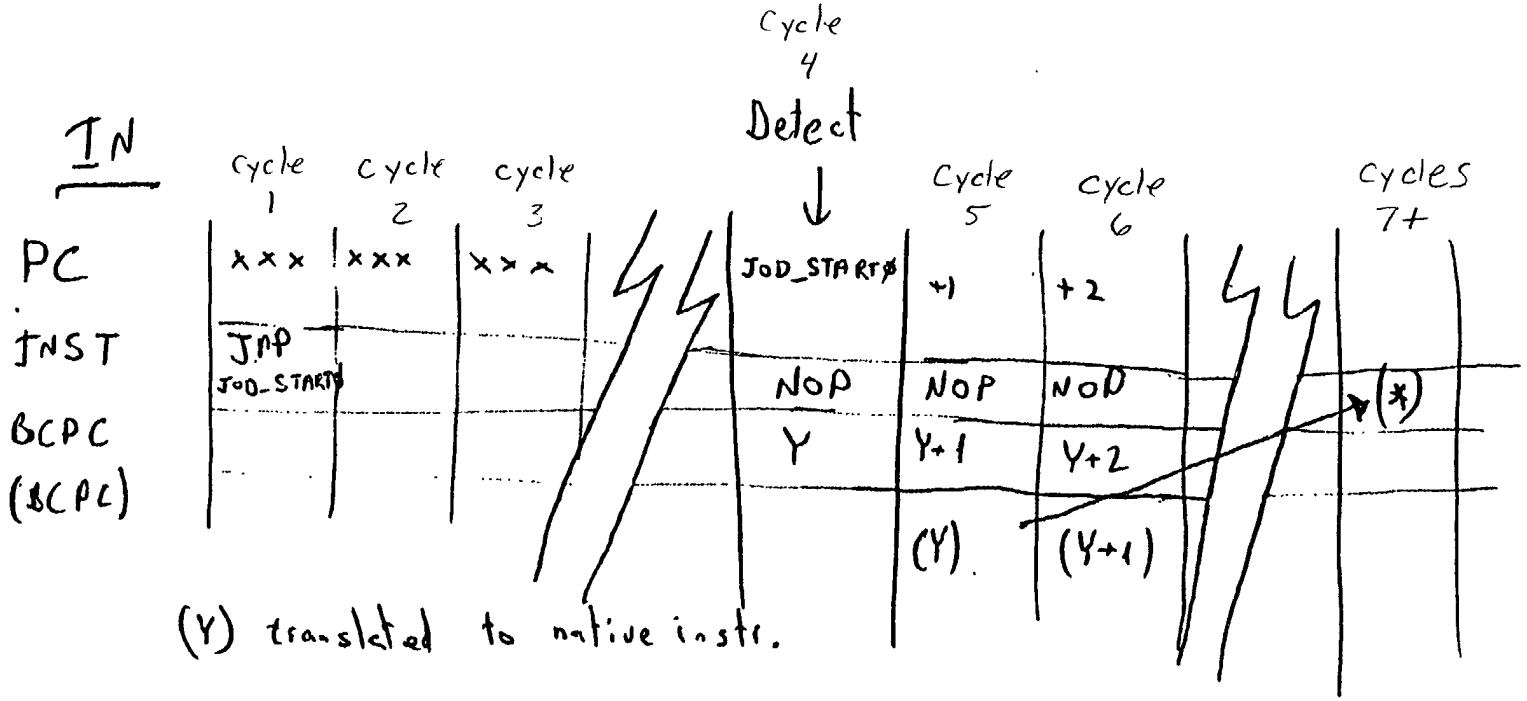

IN
Cycle 1
Cycle 2
Cycle 3
Cycle 4
Detect
Cycle 5
Cycle 6
Cycles 7+
PC
INST
BCPC
(BCPC)
xxx
xxx
xxx
JAP
JOD_START$
JOD_START$
+1
+2
NOP
NOP
NOP
Y
Y+1
Y+2
(Y)
(Y+1)
(*)
(Y) translated to native instr.

FIG. 9a

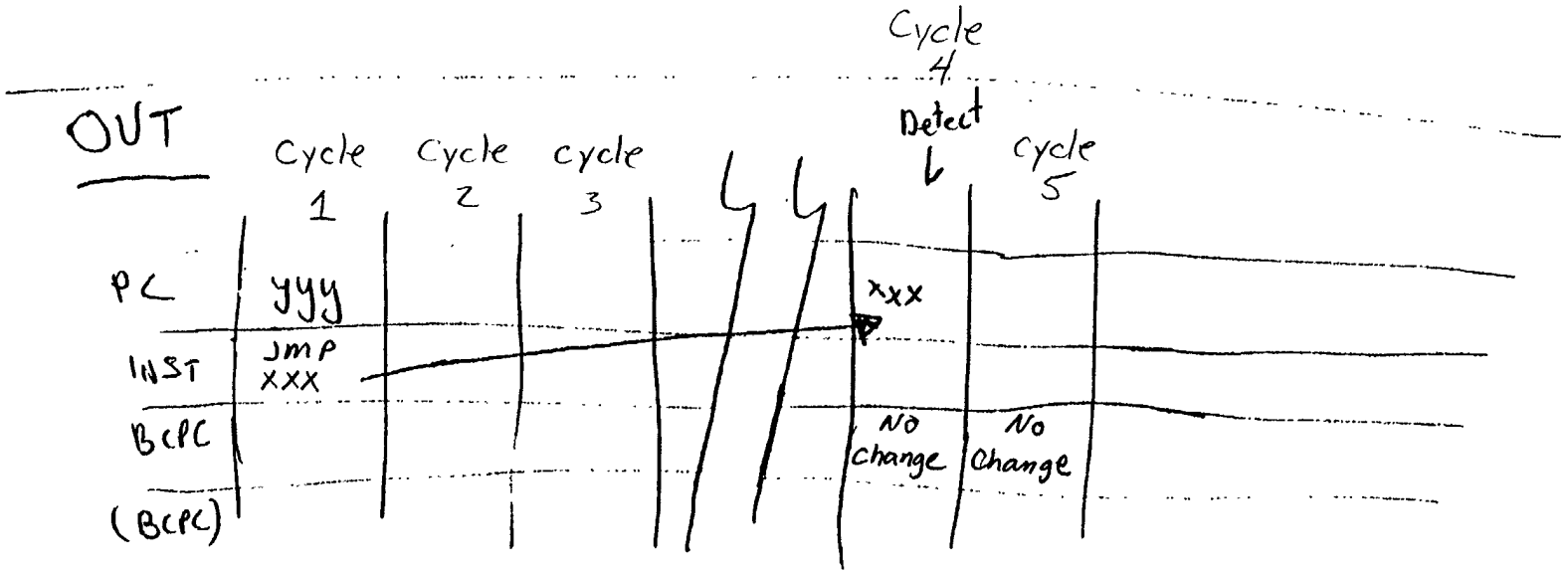

OUT
Cycle 1
Cycle 2
cycle 3
Cycle 4
Detect
cycle 5
PC
yyy
xxx
INST
JMP xxx
BCPC
(BCPC)
No change
No change

FIG. 9b

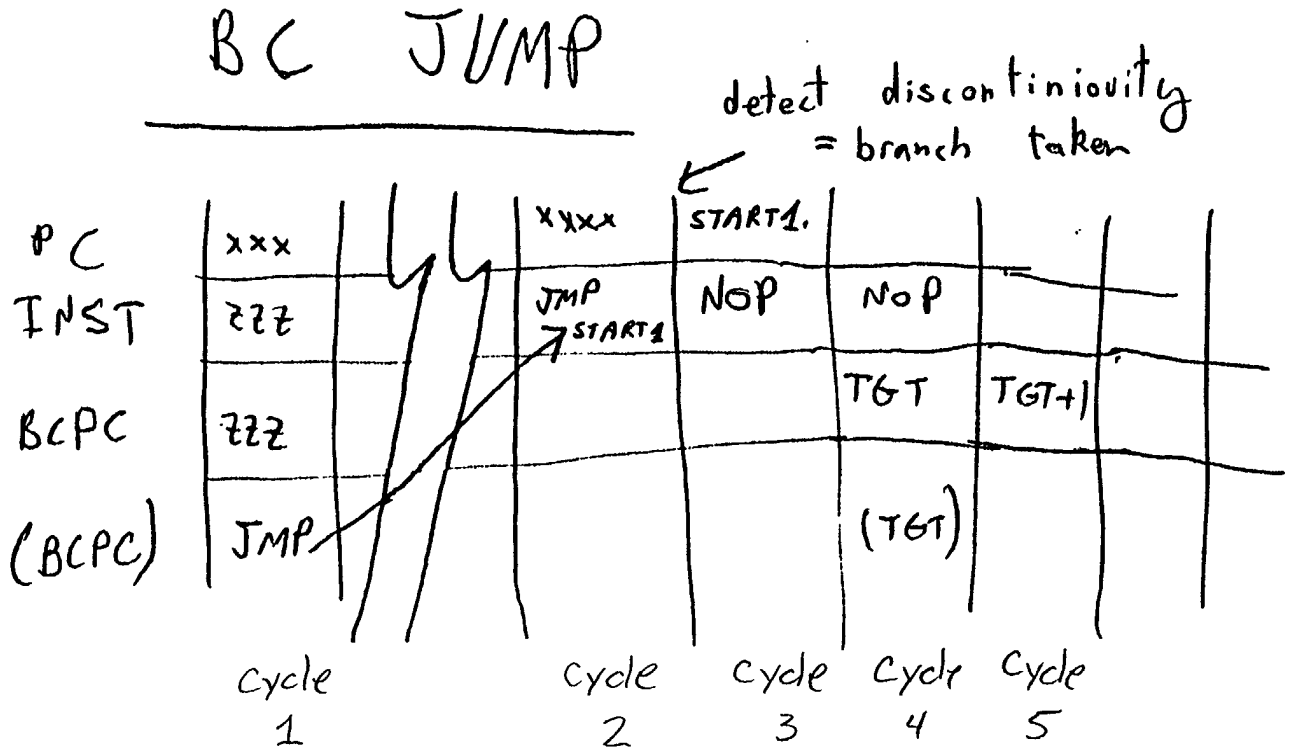

BC JUMP
detect discontiniouity
= branch taken
PC
INST
BCPC
(BCPC)
xxx
zzz
zzz
JMP
xxxx
JMP START1
START1.
NOP
NOP
TGT
TGT+1
(TGT)
Cycle 1
Cycle 2
Cycle 3
Cycle 4
Cycle 5

FIG. 10

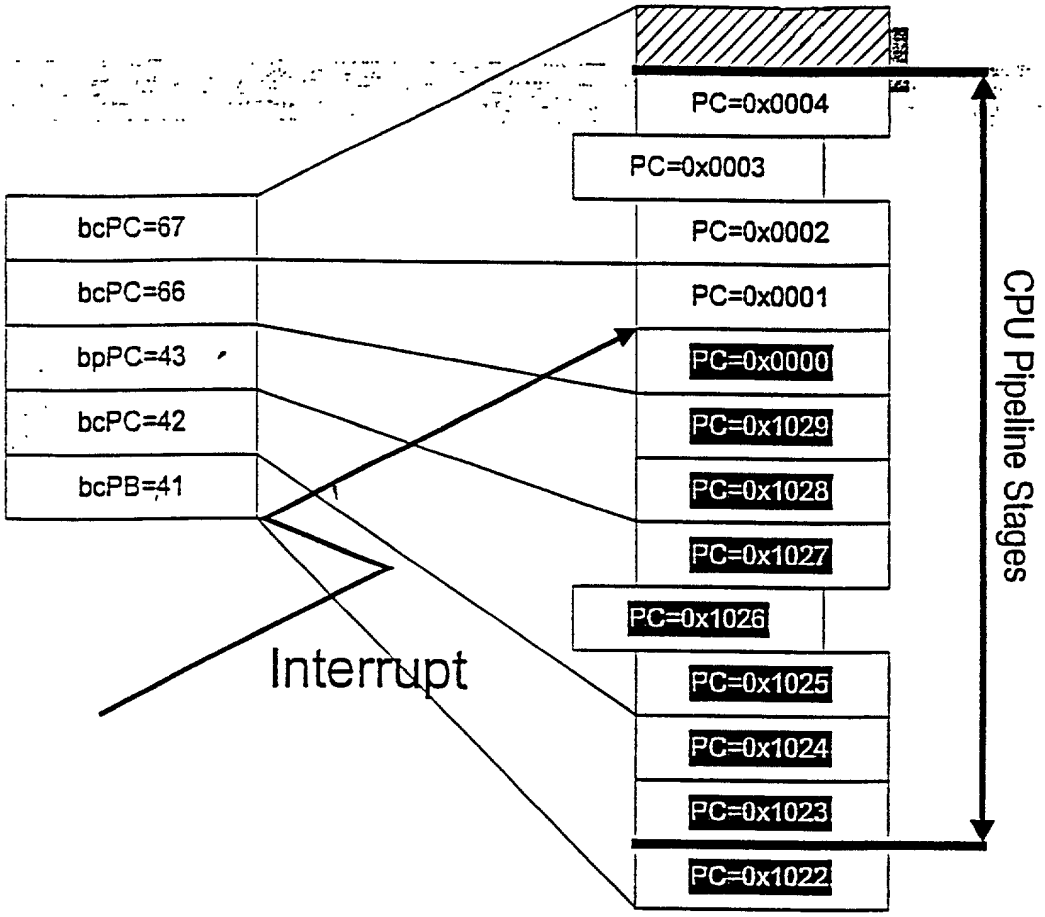

PC=0x0004
PC=0x0003
PC=0x0002
PC=0x0001
PC=0x0000
PC=0x1029
PC=0x1028
PC=0x1027
PC=0x1026
PC=0x1025
PC=0x1024
PC=0x1023
PC=0x1022
bcPC=67
bcPC=66
bpPC=43
bcPC=42
bcPB=41
Interrupt
CPU Pipeline Stages

FIG. 11